

L-Graphs and Monotone L-Graphs

Abu Reyan Ahmed, Felice De Luca, Sabin Devkota, Alon Efrat, Md Iqbal Hossain,
Stephen Kobourov, Jixian Li, Sammi Abida Salma, and Eric Welch

Department of Computer Science
University of Arizona

Abstract. An L-segment consists of a horizontal and a vertical straight line which form an L. In an L-embedding of a graph, each vertex is represented by an L-segment, and two segments intersect each other if and only if the corresponding vertices are adjacent in the graph. If the corner of each L-segment in an L-embedding lies on a straight line, we call it a monotone L-embedding. In this paper we give a full characterization of monotone L-embeddings by introducing a new class of graphs which we call “non-jumping” graphs. We show that a graph admits a monotone L-embedding if and only if the graph is a non-jumping graph. Further, we show that outerplanar graphs, convex bipartite graphs, interval graphs, 3-leaf power graphs, and complete graphs are subclasses of non-jumping graphs. Finally, we show that distance-hereditary graphs and k -leaf power graphs ($k \leq 4$) admit L-embeddings.

1 Introduction

Geometric representations of graphs have been used to reveal intriguing connections between the continuous world of geometry and the discrete world of combinatorial structures. Having a geometric representation is much more than just a way to display a graph, as it reveals underlying structures that can often be described only using geometry. A good geometric representation of a graph also leads to algorithmic solutions for purely graph-theoretic questions that, on the surface, do not seem to have anything to do with geometry. Examples of this include rubber band representations in planarity testing [20], circle-contact representations in balanced graph partitioning and approximating optimal bisection [29], volume-respecting embeddings in approximation algorithms for graph bandwidth [15], and orthogonal representations in algorithms for graph connectivity and graph coloring [21].

In an intersection representation of a graph, vertices are geometric objects (e.g., curves) and edges are realized by intersections (e.g., curve crossings). Among the most general types of intersection graphs are *string-graphs*, or graphs that admit a *string representation*, in which vertices are represented by arbitrary curves in the plane; see Fig. 1(a-b). String-graphs find a practical application in the modeling of integrated thin film RC circuits, where some pairs of conductors in a circuit can cross [28]. The class of k -*string-graphs* contains the graphs that have a string representation with at most k intersections between two strings, where $k \geq 0$. Not every graph is a string-graph; for instance, the full subdivision graph of the graph K_5 does not have a string representation; see Fig. 1(f).

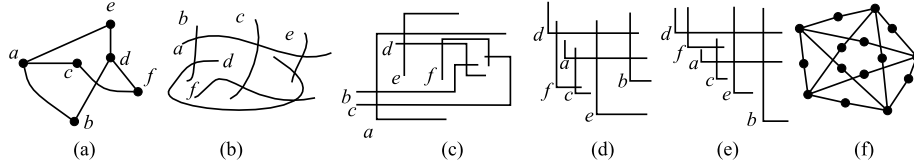


Fig. 1. (a) A graph G , (b) a string representation of G , and (c) a B_2 -VPG representation of G . (d) an L-embedding of G , (e) a monotone L-embedding of G , and (f) a graph that is not an L-graph.

Planar graphs are known to be 1-string graphs [8,9,14]. Chalopin and Gonçalves strengthen this result by proving a conjecture of Scheinerman [27] that every planar graph has a *segment representation* [10], where the segments have arbitrary slopes and intersect at arbitrary angles. The class of segment (*SEG*) graphs is included in the class of 1-string-graphs. The recognition of string-graphs is NP-hard [19,22].

Another widely-studied class of graphs is the *Vertex Path Grid (VPG)* class, introduced by Asinowski *et al.* [1,2]. The class of k -Bends VPG (B_k -VPG) graphs restricts the number of bends of the orthogonal paths to k , with $k \geq 0$; see Fig. 1(c). The class of B_k -VPG-graphs is equivalent to the class of string-graphs [1,2]. Chaplick *et al.* [11] showed that for every fixed k , the recognition of B_k -VPG-graph is NP-complete even when the input graph is given by a B_{k+1} -VPG representation. The B_k -VPG representation is related to the *edge intersection graphs of paths in a grid (EPG-graphs)* introduced by Golumbic *et al.* [18]. In an EPG representation, the vertices are represented as paths on a grid, and two vertices are adjacent if and only if their corresponding paths share a grid edge. Pergel and Rzażewski [25] proved that is NP-complete to recognize 2-bend-EPG-graphs.

The study of B_k -VPG graphs is motivated by practical applications in circuit layouts [7,23]. In the knock-knee layout model, the layout may have multiple layers, and on each layer, the vertex intersection graph of paths on a grid is an independent set. This corresponds to a graph coloring problem, and the minimum coloring problem of VPG-graphs defines the knock-knee multiple layout with minimum number of layers. This model is used by Asinowski *et al.* [2], who studied VPG-graphs and showed that interval graphs and trees are both subfamilies of B_0 -VPG, and that *circle graphs* are contained in the class B_1 -VPG (where circle graphs are string graphs in which the strings are chords of a circle). Since the problem of coloring a circle graph is NP-complete [17], it follows that the coloring problem is also NP-complete for B_1 -VPG-graphs. Asinowski *et al.* [2] proved that the coloring problem remains NP-complete even for B_0 -VPG-graphs.

The class B_k -VPG contains all planar graphs, and a central question is how small k can be. Asinowski *et al.* [2] showed that every planar graph is a B_3 -VPG-graph and Chaplick and Ueckerdt [12] showed that every planar graph is a B_2 -VPG-graph.

In B_1 -VPG-graphs, four possible L-shapes, L, J, Γ and \mathcal{L} , are may be used to represent vertices. In an L-graph, the vertices are represented with only one of these L-shapes. Biedl and Derka [5,6] show that series-parallel graphs, Halin-graphs, and outerplanar graphs are L-graphs. Felsner *et al.* [16] show that every planar 3-tree is an L-graph, and

that full subdivisions of planar graphs and line graphs of planar graphs are L-graphs. On the other hand, full subdivisions of non-planar graphs are not L-graphs [28].

In the rest of this paper we restrict our focus to graphs that have an L-representation, which we refer to as L-graphs. Formally, an L-segment consists of a horizontal and a vertical straight line which form an L. An *L-embedding* is a drawing of G in which each vertex is drawn as an L-segment, and two segments intersect each other if and only if the corresponding vertices are adjacent in the graph. G is an *L-graph* if it admits an L-embedding. If the corner of each L-segment in an L-embedding lies on a straight line, then it is called a *monotone L-embedding*. A graph is called a *monotone L-graph* if it admits a monotone L-embedding.

Our contributions: We study L-graphs and monotone L-graphs and summarize our results as follows:

- We introduce a new class of graphs which we call “non-jumping graph” and a new vertex labeling which we call “non-jumping labeling.”
- We give a full characterization of monotone L-graphs by showing that a graph admits a monotone L-embedding if and only if the graph is a non-jumping graph.
- We show that given a graph G on n vertices and m edges with labeling γ , there is an $O(n \log n + m)$ time algorithm to determine whether γ is a non-jumping labeling.
- We show that outerplanar graphs, convex bipartite graphs, interval graphs, and complete graphs are subclasses of non-jumping graphs.
- We show that distance-hereditary graphs and k -leaf power graphs ($k \leq 4$) admit L-embeddings.

The rest of the paper is organized as follows. Section 2 defines some preliminary graph-theoretic terminology. In Section 3, we define a “non-jumping graph” and show that (bull, dart, gem)-free chordal graphs, interval graphs, outerplanar graphs, complete graphs, and convex bipartite graphs are non-jumping graphs. We also provide an algorithm to compute a monotone L-embedding of a non-jumping graph, and describe some of the properties of non-jumping graphs. In Section 4, we show that distance-hereditary graphs and 4-leaf power graphs admit L-embeddings. We conclude the paper with some open problems.

2 Preliminaries

In this section we introduce several definitions. For graph-theoretic definitions not described here, see [24].

Let $G = (V, E)$ be a graph with a set of vertices V and a set of edges E . We say that G is *connected* if there is a path between every pair of vertices in V . A *cycle* of G is a path in which every vertex is reachable from itself. G is a *tree* if it does not contain any cycles. G is *planar* if it can be embedded in the plane without edge crossings, and *outerplanar* if it has a planar drawing in which all vertices of G are placed on the outer face of the drawing. G is *bipartite* if its vertices can be partitioned into sets R and B such that every edge connects a vertex in R to one in B . The set of neighbors of v is denoted by $N(v)$. If a bijective mapping $f : B \rightarrow \{1, 2, \dots, |B|\}$ exists such that for all $r \in R$, and for any two vertices $x, y \in N(r)$, there does not exist a vertex $z \in B \setminus N(r)$

such that $f(x) < f(z) < f(y)$, then G is called a *convex bipartite* graph. G is called an *interval graph*, and a set of intervals S is called an *interval representation* of G , if there exists a one-to-one correspondence between vertices of G and intervals in S , such that u and v are adjacent in G , if and only if, their corresponding intervals intersect.

Let $G' = (V', E')$ be a graph such that $V' \subseteq V$ and $E' \subseteq E$. Then G' is called a *subgraph* of G . The subgraph G' is an *induced subgraph* of G if E' consists of all the edges in E that have both endpoints in V' . G is a *distance-hereditary graph* if and only if for every pair of vertices u, v all induced path between u and v have the same length.

G is a *k-leaf power* graph if there is a tree T whose leaves correspond to the vertices of G in such a way that two vertices are adjacent in G precisely when their distance in T is at most k . We say that G is a *leaf power graph* if it is a k -leaf power for some k .

A vertex $u \in V$ is a *pendant* vertex if it has degree 1. For two vertices $u, v \in V$, if u and v are neighbors and $N(u) \setminus \{v\} = N(v) \setminus \{u\}$, then u and v are called *true twins*. If u and v are not neighbors and $N(u) = N(v)$, we say that u and v are *false twins*.

A vertex v is called *simplicial* in G if the subgraph of G induced by the vertex set $\{v\} \cup N(v)$ is a complete graph. An ordering of $\{v_1, v_2, \dots, v_n\}$ is a *perfect elimination ordering* of G if each v_i is simplicial in the subgraph induced by the vertices $\{v_1, v_2, \dots, v_i\}$. G is a *chordal* graph if it has a perfect elimination ordering.

An *L-segment* consists of a horizontal and a vertical straight-line segment which together look exactly like an L, with no rotation. Let \mathbb{L} be an L-embedding of G and v be a vertex of G . We denote the corresponding L-segment of v in \mathbb{L} by $L(v)$. The L-segment is defined by its corner position, the height of its vertical and the width of its horizontal line segments, denoted by $(v.x, v.y)$, $v.h$, and $v.w$, respectively. Let $L(u)$ and $L(v)$ be two L-segments in \mathbb{L} . The segments $L(u)$ and $L(v)$ might cross each other multiple times in case of overlapping horizontal or vertical segments. In this paper we consider only L-embeddings with single crossings, so that if $(u, v) \in E$ then either the horizontal segment of $L(v)$ crosses the vertical segment of $L(u)$, or the vertical segment of $L(v)$ crosses the horizontal segment of $L(u)$.

3 Non-jumping graphs

In this section we give a formal definition of a non-jumping graph. Then we show that several classes of graphs are non-jumping graphs. Before we define non-jumping graphs, we must first define a non-jumping labeling of a graph $G = (V, E)$. A *non-jumping labeling* of G is a vertex labeling v_1, v_2, \dots, v_n such that if $(v_i, v_k), (v_j, v_l) \in E$ and $i < j < k < l$, then $(v_j, v_k) \in E$. Figure 2(a) provides an example of a non-jumping labeling. If G admits a non-jumping labeling, then we say that G is a *non-jumping graph*; if G has no non-jumping labeling then G is called a *jumping graph*. If a vertex labeling contains a vertex v_j such that $(v_i, v_k), (v_j, v_l) \in E$ but $(v_j, v_k) \notin E$ (where $i < j < k < l$), then v_j is called a *jumping vertex* for v_i, v_k , and v_l . For example, the vertex v_3 is a jumping vertex in the graph shown in Fig. 2(c). Clearly, a non-jumping labeling does not contain any jumping vertex.

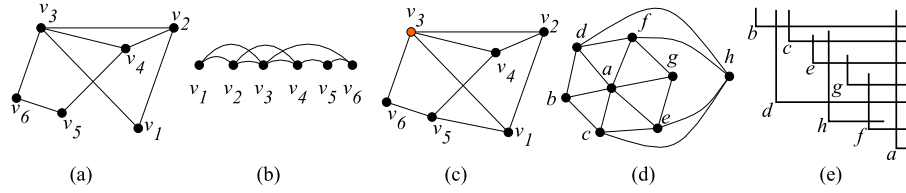


Fig. 2. (a) A non-jumping labeling of a graph G , (b) an ordering $\gamma = \{v_1, v_2, v_3, v_4, v_5\}$ of G , (c) a jumping vertex v_3 , (d) a jumping graph G' , and (e) an L-embedding of G' .

3.1 Families of non-jumping graphs

One can easily verify that paths, cycles, and complete graphs are non-jumping graphs. In this section, we describe several other types of graphs can be classified as non-jumping graphs. We begin with outerplanar graphs. We begin with outerplanar graphs.

Theorem 1. *Let G be an outerplanar graph. Then G is a non-jumping graph, and a non-jumping labeling of G can be found in linear time.*

Proof. Every outerplanar graph admits a one page book embedding [4] which can be found in linear time. In a one page book embedding of a graph, we place each vertex of the graph on the spine of the book and each edge can be drawn on one page without edge crossing. If we consider the sequence of vertices as a labeling of a one page book embedding, there is no jumping vertex because there is no pair of edges $(v_i, v_k), (v_j, v_l) \in E$ where $i < j < k < l$. \square

We next show that (bull, dart, gem)-free chordal graphs are non-jumping graphs. The bull, dart and gem are shown in the Fig. 3(a). Before the proof, we define a few terms as follows: Let T be a tree, and v be a vertex of T . We denote the subtree of T rooted at v by T_v . We denote the parent of v by v' , and parent of v' by v'' . A vertex u is said to be an *uncle* of v if $u' = v''$.

Theorem 2. *Every (bull, dart, gem)-free chordal graph is a non-jumping graph.*

Proof. Let $G = (V, E)$ be a (bull, dart, gem)-free chordal graph of n vertices. Then there is a tree T whose leaves correspond to the vertices of G such that two vertices are adjacent in G precisely when their distance in T is at most three [26]; see Fig. 3(b-c). Hence G is a 3-leaf power graph of T . We use the notation \bar{v} to indicate that the leaf v of T corresponds to the vertex \bar{v} in G . Let \bar{u} and \bar{v} be vertices of G .

Since G is a 3-leaf power graph of T , $(\bar{u}, \bar{v}) \in E$ if and only if u and v are siblings, or u is an uncle of v , or v is an uncle of u . We find an ordering of vertices of G using T as follows: We first root T at a non-leaf vertex x of T . We then sort each subtree of T rooted at each vertex in counterclockwise order, according to depth in ascending order.

Let $\gamma' = \{v_1, v_2, v_3, \dots, v_n\}$ be the ordering of the leaves taken from the counterclockwise DFS traversal on T starting from x . We now prove that $\gamma = \{\bar{v}_1, \bar{v}_2, \bar{v}_3, \dots, \bar{v}_n\}$ is a non-jumping labeling of G by supposing that $(\bar{v}_i, \bar{v}_k), (\bar{v}_j, \bar{v}_l) \in E$ with $i < j < k < l$, and showing that $(\bar{v}_j, \bar{v}_k) \in E$.

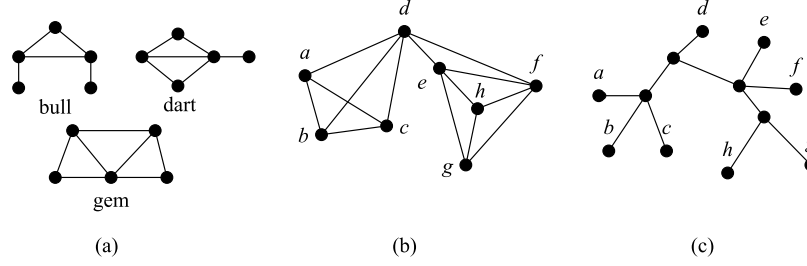


Fig. 3. (a) A bull, a dart, a gem. (b) A 3-leaf power graph G , and (c) the 3-leaf power tree of G .

It is easy to see that $(\overline{v_i}, \overline{v_k}) \in E$ if and only if v_i and v_k are siblings, or v_i is an uncle of v_k . Since we sorted the vertices by their depth before taking the ordering, and since $i < k$, v_k can not be an uncle of v_i . Thus, we have two cases to consider:

Case 1: v_i and v_k are siblings. Since the order was taken from DFS traversal, v_i, v_{i+1}, \dots, v_k are siblings. So, v_j and v_k are siblings, and we have $(\overline{v_j}, \overline{v_k}) \in E$.

Case 2: v_i is an uncle of v_k . If v_i and v_j are siblings, then $(\overline{v_j}, \overline{v_k}) \in E$, because the distance between v_i and v_k and the distance between v_j and v_k are the same. Otherwise, we can prove that v_j and v_k are siblings. Suppose that v_j and v_k are not siblings. Then $v_j \in T_o$, where o is a non-leaf child of v'_i that was encountered before v'_k in the traversal. Thus, the path between v_j and v_l contains v'_i due to the ordering of the vertices and the positions of i, j, k , and l ; see Fig. 11. Now, the distance between v'_i and v_j is at least 2, and the distance between v'_i and v_l is at least 2. This means that $(\overline{v_j}, \overline{v_l}) \notin E$, which is not true. The contradiction shows that v_j and v_k must be siblings and $(\overline{v_j}, \overline{v_k}) \in E$. \square

We now show that every interval graph has a non-jumping labeling.

Theorem 3. *Every interval graph is a non-jumping graph.*

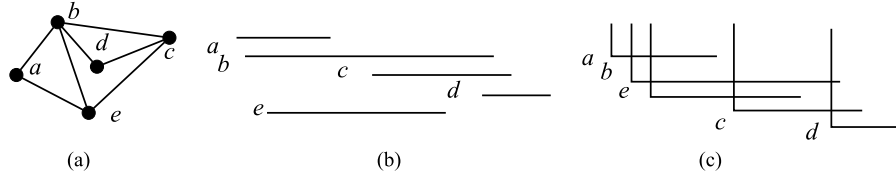


Fig. 4. (a) A graph G , (b) an interval representation of G , and (c) an L-embedding of G .

Proof. Let G be an interval graph of n vertices. Let $S = \{|a_1, b_1|, |a_2, b_2|, \dots, |a_n, b_n|\}$ be an interval representation of G . We denote the endpoints a_i and b_i of the interval corresponding to v_i by $v_i(a)$ and $v_i(b)$, respectively. Note that $v_i(a) < v_i(b)$; see Fig. 4. Let $\gamma = \{v_1, v_2, \dots, v_n\}$ be an ordering of vertices of G in non-decreasing order of $v_i(a)$ ($1 \leq i \leq n$). If $i < j$ then $v_i(a) < v_j(a)$. We now prove that γ is a non-jumping labeling of G . By way of contradiction, assume that γ is a jumping labeling. Then γ contains a jumping vertex v_j . By definition, there exists edges $(v_i, v_k), (v_j, v_l) \in E$ with

$i < j < k < l$ such that $(v_j, v_k) \notin E$. This means $v_i(b) > v_k(a)$ and $v_j(b) > v_l(a)$. By construction, $v_j(b) > v_k(a)$, since $v_k(a) < v_l(a)$ and $v_l(a) < v_j(b)$. Since $v_j(b) > v_k(a)$ and $v_j(a) < v_k(a)$, we have $(v_j, v_k) \in E$, a contradiction. \square

Theorem 4. *Every convex bipartite graph is a non-jumping graph.*

Proof. Let $G = (R \cup B, E)$ be a convex bipartite graph with $V(G) = R \cup B$, where $R \cap B = \emptyset$. Without loss of generality, suppose that G is convex over B . Then there exists a bijective mapping $f : B \rightarrow \{1, 2, \dots, |B|\}$ such that for all $v \in R$ and any two vertices $x, y \in N(v)$, there is no vertex $z \in B \setminus N(v)$ such that $f(x) < f(z) < f(y)$.

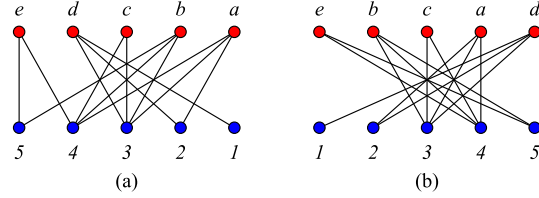


Fig. 5. (a) A convex bipartite graph G and (b) a vertex ordering of G .

For γ to be a non-jumping labeling, it must be true that for all positions $i < j < k < l$ in γ , if $(v_i, v_k) \in E$ and $(v_j, v_l) \in E$, then $(v_j, v_k) \in E$. Consider such a pair of edges $(v_i, v_k), (v_j, v_l) \in E$. Since $i < k$ and G is a bipartite graph, $v_i \in R$ and $v_k \in B$. Similarly, $v_j \in R$ and $v_l \in B$.

Because $i < j$ and the vertices in R_{sort} were ordered in non-increasing order of $s(r)$, we have $s(v_i) \geq s(v_j)$. Also, since $v_k \in N(v_i)$, we know that $f(v_k) \geq s(v_i)$. Consequently, $f(v_k) \geq s(v_i) \geq s(v_j)$.

Since $(v_j, v_l) \in E$ and G is convex on B , $N(v_j)$ must contain all vertices $v \in B$ whose mapping $f(v)$ lies in the interval $[s(v_j), f(v_l)]$. Because $f(v_k) \geq s(v_j)$ and, from the ordering in B_f , $f(v_k) < f(v_l)$, we find that $f(v_k)$ lies in the interval $[s(v_j), f(v_l)]$. Thus, $v_k \in N(v_j)$, which means that $(v_k, v_j) \in E$, as required. \square

3.2 Characterization of non-jumping graphs

The graph shown in the Fig. 2(d) is an example of jumping graph. In Theorem 5, we prove that there is no non-jumping labeling for this graph. Due to space limitations, the proof of Theorem 5 is given in the Appendix.

Theorem 5. *Not all graphs are non-jumping graphs.*

Recall that a monotone L-embedding is an L-embedding such that the corners of each L-segment are on a straight line. We can completely characterize monotone L-graphs in terms of non-jumping graphs.

Theorem 6. *A graph G admits a monotone L-embedding if and only if G is a non-jumping graph.*

We prove Theorem 6 by first showing that any non-jumping graph G admits a monotone L-embedding in Lemma 1. The converse is proven in Lemma 2.

Before we begin, we note that if a graph has a monotone L-embedding with the corners of the L's on a line that is drawn vertically or horizontally, then for any pair of vertices (v_i, v_j) , there can only be an edge (v_i, v_j) if $i + 1 = j$, i.e., the graph is a subgraph of a path. Thus, the graph is trivially non-jumping, as there cannot be any indices $i < j < k < l$ in a labeling γ such that $(v_i, v_k) \in E$ or $(v_j, v_l) \in E$. A graph with no edges is also trivially non-jumping, and admits a “degenerate” monotone L-embedding in which no L would intersect another even if their arms were extended indefinitely.

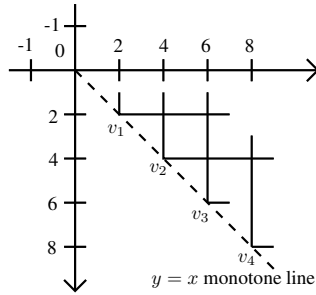


Fig. 6. Monotone positioning of the L-segments corresponding to v_1, \dots, v_4 on the line $y = x$.
 indicated, every monotone L-embedding will have its corners aligned on the line $y = x$ in this way.

For convenience, we define a coordinate system over the quarter-plane \mathbb{R}^2 beginning with $(0, 0)$ in the top-left corner, and x - and y -coordinates increasing to the right and downward respectively. This choice of coordinate system will allow us to construct a monotone L-embedding so that the corner of every L lies on the line $y = x$. Moreover, any non-trivial¹ monotone L-embedding can be expanded (see Lemma 3), translated, and rescaled to create an equivalent embedding with the corners of each L on this line. Note that once we have a drawing with the corners of each L line arranged on $y = x$, we can perform arbitrary affine transformations on the coordinate system without rotating any of the L's themselves.

For the rest of the paper, unless otherwise indicated, every monotone L-embedding will have its corners aligned on the line $y = x$ in this way.

Lemma 1. *Let G be a non-jumping graph of n vertices and m edges. Then G admits monotone L-embedding on a grid of size $O(n) \times O(n)$, and this embedding can be computed in $O(n + m)$ time.*

Proof. Let G be a non-jumping graph of n vertices and $\gamma = \{v_1, v_2, \dots, v_n\}$ be a non-jumping labeling of G . If there are edges $(v_i, v_k), (v_j, v_l) \in E$ such that $i < j < k < l$, then $(v_j, v_k) \in E$.

We now construct an L-monotone drawing for G using the coordinate system given above. Let $L(v)$ be the L-drawing of vertex v . Then $(v.x, v.y)$ is the corner of $L(v)$ and the horizontal and vertical arms of $L(v)$ have lengths $v.w$ and $v.h$ respectively.

For each $v_j \in V$, let $v_j.x = v_j.y = 2j$; this places all corners on the line $y = x$. Also, for each v_j , if there exists an index $i < j$ such that some $(v_i, v_j) \in E$, then for

¹ By non-trivial, we mean a monotone L-embedding with at least one intersection, and with L's that are not aligned horizontally or vertically.

$a = \min\{i \mid i < j \text{ and } (v_i, v_j) \in E\}$, define $v_j.h = 2|j - a| + 1$. If there is no such index i , then let $v_j.h = 1$. Similarly, if there is some index $k > j$ such that $(v_j, v_k) \in E$, then let $b = \max\{k \mid k > j \text{ and } (v_j, v_k) \in E\}$ and define $v_j.w = 2|j - b| + 1$; otherwise, let $v_j.w = 1$; see Fig. 6.

To see that this is a valid L-monotone drawing, first recall that the corners of all the L's are on the diagonal line $y = x$. Also note that for each index $a < b$, $v_a.x < v_b.x$ and $v_a.y < v_b.y$. We must show that for indices a and b with $(v_a, v_b) \in E$, $L(v_a)$ and $L(v_b)$ intersect. Without loss of generality, suppose $a < b$. Then, we must show that: $|v_a.x - v_b.x| < v_a.w$ and $|v_a.y - v_b.y| < v_b.h$; Finally, we must show that for $(v_j, v_k) \notin E$ with $j < k$, either $|v_j.x - v_k.x| > v_j.w$ or $|v_j.y - v_k.y| > v_k.h$.

It is clear from the computation of the width and height of each L that whenever $(v_a, v_b) \in E$, $L(v_a)$ and $L(v_b)$ intersect as described above.

Now let v_j and v_k be vertices in G such that $(v_j, v_k) \notin E$, with $j < k$. Suppose for a contradiction that $L(v_j)$ and $L(v_k)$ intersect; that is, $|v_j.x - v_k.x| < v_j.w$ and $|v_j.y - v_k.y| < v_k.h$. Since v_j and v_k are not adjacent, by the construction of $v_j.w$, there must be some vertex v_l such that $l > k$ and

$$v_j.w = |v_j.x - v_l.x| + 1 > |v_j.x - v_k.x|$$

Similarly, by the construction of $v_l.h$, there must be some vertex v_i such that $i < j$ and

$$v_k.h = |v_k.y - v_i.y| + 1 > |v_j.y - v_k.y|$$

We now have $i < j < k < l$ and the two edges $(v_i, v_k), (v_j, v_l) \in E$. Since the indices i, j, k, l are taken from a non-jumping labeling of G , we must have $(v_j, v_k) \in E$, a contradiction.

The entire drawing is contained in a rectangle of dimensions $2n \times 2n$. To see this, note that no corner of any L-segment will be placed to the left of the line $x = 2$, nor below the line $y = 2n$. Also, no horizontal arm of an L will extend to the right beyond the line $x = 2n + 1$, as this is one unit to the right of $L(v_n)$, nor will any vertical arm extend above the line $y = 1$.

We can construct this drawing in $O(|V| + |E|)$ time. First, for each $v \in V$, we plot the corner of $L(v)$ at $(v.x, v.y)$, and draw its two arms with unit length. Then, for each edge $(v_i, v_j) \in E$ with $i < j$, we extend the horizontal arm of $L(v_i)$ to have length at least $2|i - j| + 1$, and extend the vertical arm of $L(v_j)$ to have length at least $2|i - j| + 1$. \square

Lemma 2. *Let \mathbb{L} be a monotone L-embedding of a graph G . Then G is a non-jumping graph.*

Proof. Since \mathbb{L} is monotone, the corners of each $L(v)$ lie on a straight line. Let $\gamma(\mathbb{L}) = \{L(v_1), L(v_2), \dots, L(v_n)\}$ be an ordering of the L-segments according to their corner positions from left to right. If the line on which the corners of the L's lie is horizontal or vertical, then as described above, the graph is a subgraph of a path, and is trivially non-jumping. Similarly, if the corners lie on a line with negative slope, then there are no edges, so the graph is trivially non-jumping.

The remaining possibility is that the corners of the L-segments lie on a line with positive slope. In this case, for each pair of indices $a < b$, we have $v_a.x < v_b.x$ and

$v_a.y < v_a.y$. Now, $\gamma(\mathbb{L})$ gives us an ordering $\gamma = \{v_1, v_2, \dots, v_n\}$ of the corresponding vertices of G . We want to prove that γ is a non-jumping labeling of G . For any four vertices v_i, v_j, v_k, v_l with $i < j < k < l$, we must show that if $L(v_i)$ intersects $L(v_k)$ and $L(v_j)$ intersects $L(v_l)$, then $L(v_j)$ and $L(v_k)$ also intersect.

To begin, note that if $L(v_i)$ and $L(v_k)$ intersect, then $|v_i.x - v_k.x| < v_i.w$ and $|v_i.y - v_k.y| < v_k.h$. Similarly, if $L(v_j)$ and $L(v_l)$ intersect, we have $|v_j.x - v_l.x| < v_j.w$ and $|v_j.y - v_l.y| < v_l.h$. By the ordering of $\gamma(\mathbb{L})$, we have $v_i.x < v_j.x < v_k.x$ and $v_j.y < v_k.y < v_l.y$. Thus, $|v_j.x - v_k.x| < |v_j.x - v_l.x| < v_j.w$ and $|v_j.y - v_k.y| < |v_i.y - v_k.y| < v_k.h$. So $L(v_j)$ and $L(v_k)$ intersect, and $(v_j, v_k) \in E$. \square

In the proof of Theorem 5, we show that the graph in Fig. 2(d) is a jumping graph. However, it is easy to verify that the L-embedding in Fig. 2(e) is the L-embedding of the jumping graph. In Theorem 6, we showed that a graph G admits a monotone L-embedding if and only if G is a non-jumping graph. This proves the following theorem:

Theorem 7. *Not all L-graphs are monotone L-graphs.*

3.3 Recognition of a non-jumping labeling

While it is difficult to determine whether a particular graph is non-jumping, the following theorem shows that we can easily verify whether a given labeling for a graph is a non-jumping labeling.

Theorem 8. *Given a graph $G = (V, E)$ with vertex labeling $\gamma = \{v_1, v_2, \dots, v_n\}$, it can be determined in $O(|V| \log |V| + |E|)$ time whether γ is a non-jumping labeling for G .*

Proof. Using the procedure described in Lemma 1, we can construct an L-monotone embedding of a graph $G = (V, E)$ in $O(|V| + |E|)$ time given a non-jumping labeling γ . Let us call the procedure P .

From Lemma 2, we know that given any L-monotone embedding, we can construct a non-jumping labeling γ by sorting the vertices $v_i \in G(V)$ in increasing order of this corner coordinate $v_i.x$. Let us call this order V_{sort} . The γ thus constructed from V_{sort} is a non-jumping labeling.

P produces a valid L-monotone embedding if and only if the input labeling γ is non-jumping. To prove this, let us suppose we get a valid L-monotone embedding from P using a jumping labeling γ_{jump} . Let us arrange the vertices v_i in increasing order of corner coordinates $v_i.x$ to obtain V_{sort} . V_{sort} must give a non-jumping labeling. Thus, our assumption that γ_{jump} is a jumping labeling is invalid and we get a contradiction.

We can use this to test if any γ is non-jumping or not. Let the drawing produced by P using γ be $P(\gamma)$. If $P(\gamma)$ is a valid L-monotone embedding, γ must be non-jumping. A valid L-embedding has $2|V|$ line segments (one vertical and one horizontal for each L-shape). Similarly, there are $|E| + |V|$ line segment intersections (one intersection for every $(v_i, v_j) \in E$, and one intersection at the corner of each L-shape).

Using an orthogonal line segment intersection search (e.g., a sweep line algorithm as described in [13]), all intersections in $P(\gamma)$ can be listed in $O(N \log N + k)$ time,

where $N = 2|V|$ is the number of total line segments, and $k = O(N^2)$ is the number of possible intersections. It suffices to check the first $|E| + |V|$ intersections to determine if $P(\gamma)$ is a valid L-monotone embedding: if there is an unwanted intersection in the first $|E| + |V|$ intersections, then the embedding is invalid, and γ is jumping. On the other hand, if there are more than $|E| + |V|$ intersections, these additional intersections must be invalid and γ is jumping. Otherwise, γ is non-jumping.

Since only the first $k = |E| + |V|$ intersections need to be examined, we only need $O(|V| \log |V| + |E|)$ time to determine if a labeling is non-jumping or not. \square

4 Other L-graphs

In this section we prove that distance-hereditary graphs and k -leaf power graphs (for $k \leq 4$) admit L-embeddings. We begin with a lemma about transformations of L-embeddings; this result will allow us to derive new L-graphs from old ones.

Lemma 3. *Let \mathbb{L} be an L-embedding of a graph G , and $L(v)$ be the L corresponding to vertex v . Then, a valid L-embedding \mathbb{L}' of G can be constructed from \mathbb{L} by expanding an infinitesimal slice of \mathbb{L}' that is parallel to an arm of $L(v)$.*

Proof. There are four ways in which \mathbb{L} can be expanded:

- a) Expand \mathbb{L} rightward with respect to $L(v)$ as follows: For every $L(w)$ in \mathbb{L} with a corner to the right of $L(v)$, move $L(w)$ to the right by one unit; also, for every vertex $L(u)$ with a corner to the left of or vertically aligned with $L(u)$ that intersects such an $L(w)$, extend the horizontal arm of $L(u)$ to the right by one unit.
- b) Expand \mathbb{L} leftward with respect to $L(v)$ as follows: For every $L(u)$ in \mathbb{L} with a corner to the left of $L(v)$, move $L(u)$ to the left by one unit; also, if such an $L(u)$ intersects a $L(w)$ that has its corner vertically aligned with or to the right of $L(v)$, then extend the horizontal arm of $L(u)$ by one unit.
- c) Expand \mathbb{L} upward with respect to $L(v)$, by replacing the words ‘right’ and ‘left’ in the description of expanding rightward above with ‘up’ and ‘down’ respectively, and exchanging the words ‘horizontal’ and ‘vertical’.
- d) Expand \mathbb{L} downward by similarly modifying the description of a leftwards expansion.

To show that any of these operations produces another valid embedding \mathbb{L}' of G , we can simply observe that in each transformation, all intersections of L’s are preserved, and no new intersections are introduced. \square

Using this result, we find that certain modifications of an L-graph result in another L-graph.

Theorem 9. *Let G be a graph that admits an L-embedding, and G' be a graph constructed from G by adding a pendant vertex, a true twin, or a false twin in G . Then G' admits an L-embedding.*

Proof. Let \mathbb{L} be an L-embedding of G , and suppose we derive G' from G by adding a pendant vertex v to G with neighbor u . Let $L(u)$ represent u in \mathbb{L} . To create \mathbb{L}' , we must place $L(v)$ so that it intersects with $L(u)$ and no other L . To be sure there is room to do so, we first expand \mathbb{L} rightward two units, and both upward and downward one unit, with respect to $L(u)$. We then place $L(v)$ with its corner one unit to the right and one unit below the corner of $L(u)$, giving it horizontal arm length 1 and vertical arm length 2.

Suppose instead that we derive G' from G by replacing a vertex u with true twin vertices v and w so that v and w are adjacent to all the neighbors of u , and are also adjacent to one another. We construct \mathbb{L}' representing G' as follows. Replace $L(u)$ with $L(v)$, so that $L(v)$ retains all the intersections of $L(u)$. Now expand the drawing both rightward and downward one unit with respect to $L(v)$ to create room for $L(w)$. We give $L(w)$ a vertical arm length that is one greater than that of $L(v)$, and a horizontal arm length one less than that of $L(v)$ after the rightward expansion, and place its corner one unit down and to the right of the corner of $L(v)$. Thus, $L(v)$ and $L(w)$ each intersect every L-segment that $L(u)$ intersected, and also intersect each other.

If we construct G' from G by replacing a vertex u with false twin vertices v and w , we can proceed similarly. We first expand \mathbb{L} leftward and downward one unit with respect to $L(v)$. Next, we place $L(w)$ one unit down and to the left of $L(v)$, and give $L(w)$ vertical and horizontal arm lengths one unit greater than those of $L(v)$. Now $L(w)$ intersects the same L-segments that $L(v)$ intersects, but does not intersect $L(v)$ itself. \square

If G is a distance-hereditary graph, then G can be built up from a single vertex by a sequence of the following three operations: a) add a pendant vertex, b) replace any vertex with a pair of false twins, and c) replace any vertex with a pair of true twins. [3] Thus, Theorem 9 immediately yields the following corollary.

Corollary 1. *Let G be a distance-hereditary graph. Then G is an L-graph.*

It is easy to see that 1-leaf power graphs and 2-leaf power graphs are L-graphs. We can use Theorem 2 and Lemma 1 to also show that 3-leaf power graphs are L-graphs. The proof of the following theorem on 4-leaf power graphs is given in the Appendix.

Theorem 10. *Every 4-leaf power graph admits an L-embedding.*

5 Conclusions and Future Work

We have shown that several classes of graphs, such as distance-hereditary graphs and k -leaf power graphs for low values of k are L-graphs. We have also provided a complete characterization of the more restricted variant of monotone L-graphs by correspondence with the class of non-jumping graphs. This type of graph has a combinatorial description, expressed as the existence of a specific type of linear order of its vertices.

The results of our paper suggest several open problems: What is the complexity of determining whether a given graph G is a non-jumping? Are all planar graphs L-graphs? Are k -leaf power graphs L-graphs for $k > 4$? Our future work will investigate these questions.

References

1. Asinowski, A., Cohen, E., Golumbic, M.C., Limouzy, V., Lipshteyn, M., Stern, M.: String graphs of k -bend paths on a grid. *Electronic Notes in Discrete Mathematics* 37, 141 – 146 (2011), <http://www.sciencedirect.com/science/article/pii/S1571065311000266>
2. Asinowski, A., Cohen, E., Golumbic, M.C., Limouzy, V., Lipshteyn, M., Stern, M.: Vertex intersection graphs of paths on a grid. *J. Graph Algorithms Appl.* 16(2), 129–150 (2012)
3. Bandelt, H.J., Mulder, H.M.: Distance-hereditary graphs. *Journal of Combinatorial Theory, Series B* 41(2), 182 – 208 (1986), <http://www.sciencedirect.com/science/article/pii/0095895686900432>
4. Bernhart, F., Kainen, P.C.: The book thickness of a graph. *Journal of Combinatorial Theory, Series B* 27(3), 320 – 331 (1979)
5. Biedl, T.C., Derka, M.: 1-string B_1 -vpg representations of planar partial $\mathcal{S}\mathcal{S}$ -trees and some subclasses. *CoRR* abs/1506.07246 (2015), <http://arxiv.org/abs/1506.07246>
6. Biedl, T.C., Derka, M.: Order-preserving 1-string representations of planar graphs. *CoRR* abs/1609.08132 (2016), <http://arxiv.org/abs/1609.08132>
7. Brady, M.L., Sarrafzadeh, M.: Stretching a knock-knee layout for multilayer wiring. *IEEE Transactions on Computers* 39(1), 148–151 (Jan 1990)
8. Chalopin, J., Gonçalves, D., Ochem, P.: Planar graphs are in 1-string. In: *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 609–617. *SODA '07*, Society for Industrial and Applied Mathematics (2007), <http://dl.acm.org/citation.cfm?id=1283383.1283449>
9. Chalopin, J., Gonçalves, D., Ochem, P.: Planar graphs have 1-string representations. *Discrete & Computational Geometry* 43(3), 626–647 (2010), <http://dx.doi.org/10.1007/s00454-009-9196-9>
10. Chalopin, J., Gonçalves, D.: Every planar graph is the intersection graph of segments in the plane: Extended abstract. In: *Proceedings of the Forty-first Annual ACM Symposium on Theory of Computing*, pp. 631–638. *STOC '09*, ACM, New York, NY, USA (2009), <http://doi.acm.org/10.1145/1536414.1536500>
11. Chaplick, S., Jelínek, V., Kratochvíl, J., Vyskočil, T.: Bend-Bounded Path Intersection Graphs: Sausages, Noodles, and Waffles on a Grill, pp. 274–285. *Springer Berlin Heidelberg*, Berlin, Heidelberg (2012), http://dx.doi.org/10.1007/978-3-642-34611-8_28
12. Chaplick, S., Ueckerdt, T.: Planar graphs as vpg-graphs. In: *International Symposium on Graph Drawing*, pp. 174–186. *Springer* (2012)
13. Chazelle, B., Edelsbrunner, H., Guibas, L.J., Sharir, M.: Algorithms for bichromatic line-segment problems and polyhedral terrains. *Algorithmica* 11(2), 116–132 (1994)
14. Ehrlich, G., Even, S., Tarjan, R.: Intersection graphs of curves in the plane. *Journal of Combinatorial Theory, Series B* 21(1), 8 – 20 (1976), <http://www.sciencedirect.com/science/article/pii/0095895676900228>
15. Feige, U.: Approximating the bandwidth via volume respecting embeddings (extended abstract). In: *30th ACM Symposium on Theory of Computing (STOC)*, pp. 90–99 (1998), <http://doi.acm.org/10.1145/276698.276716>
16. Felsner, S., Knauer, K., Mertzi, G.B., Ueckerdt, T.: Intersection graphs of l-shapes and segments in the plane. *Discrete Applied Mathematics* 206, 48 – 55 (2016), <http://www.sciencedirect.com/science/article/pii/S0166218X16300154>
17. Garey, M.R., Johnson, D.S., Miller, G.L., Papadimitriou, C.H.: The complexity of coloring circular arcs and chords. *SIAM Journal on Algebraic Discrete Methods* 1(2), 216–227 (1980), <http://dx.doi.org/10.1137/0601025>

18. Golumbic, M.C., Lipshteyn, M., Stern, M.: Edge intersection graphs of single bend paths on a grid. *Networks* 54(3), 130–138 (2009), <http://dx.doi.org/10.1002/net.20305>
19. Kratochvíl, J.: String graphs. ii. recognizing string graphs is np-hard. *Journal of Combinatorial Theory, Series B* 52(1), 67 – 78 (1991), <http://www.sciencedirect.com/science/article/pii/009589569190091W>
20. Linial, N., Lovasz, L., Wigderson, A.: Rubber bands, convex embeddings and graph connectivity. *Combinatorica* 8(1), 91–102 (1988)
21. Lovász, L.: *Geometric representations of graphs*. Bolyai Society-Springer-Verlag (1999)
22. Middendorf, M., Pfeiffer, F.: Weakly transitive orientations, hasse diagrams and string graphs. *Discrete Mathematics* 111, 393 – 400 (1993), <http://www.sciencedirect.com/science/article/pii/0012365X9390176T>
23. Molitor, P.: A survey on wiring. *J. Inf. Process. Cybern.* 27(1), 3–19 (Apr 1991), <http://dl.acm.org/citation.cfm?id=108118.108119>
24. Nishizeki, T., Rahman, M.S.: *Planar graph drawing*, vol. 12. World Scientific Publishing Co Inc (2004)
25. Pergel, M., Rzażewski, P.: On Edge Intersection Graphs of Paths with 2 Bends, pp. 207–219. Springer Berlin Heidelberg, Berlin, Heidelberg (2016), http://dx.doi.org/10.1007/978-3-662-53536-3_18
26. Rautenbach, D.: Some remarks about leaf roots. *Discrete Mathematics* 306(13), 1456 – 1461 (2006), <http://www.sciencedirect.com/science/article/pii/S0012365X06002032>
27. Scheinerman, E.R.: Intersection classes and multiple intersection parameters of graphs. In: PhD thesis. Princeton University (1984)
28. Sinden, F.W.: Topology of thin film rc circuits. *The Bell System Technical Journal* 45(9), 1639–1662 (1966)
29. Spielman, D.A., Teng, S.H.: Spectral partitioning works: Planar graphs and finite element meshes. In: 37th Symposium on Foundations of Computer Science (FOCS). pp. 96–105 (1996)

Appendix

Proof of Theorem 5

Theorem. *Not all graphs are non-jumping graphs.*

We prove this theorem by showing that the graph depicted in Fig. 2(d) is a non-jumping graph. We tested this graph with a computer program finding that all the possible $8!$ labeling are jumping. In this mathematical proof we use patterns that can occur in the labeling process to show that however a labeling is chosen, it is jumping.

Before beginning the formal proof, we define the notation depicted in Fig. 7, to make easier the identification of jumping patterns. Specifically, we provide each vertex with a name X_y^z where $X \in \{I, II, III\}$ depending on its connection with other vertices: each II is connected with two I 's, one II , and one III ; each III is connected with one I and two II 's; and the I 's are the remaining vertices. Next, we have $y \in \{l, r\}$ if $X \in \{II, III\}$ and $y \in \{in, out\}$ if $X = \{I\}$, such that III_l (III_r) is connected with two II_l 's (II_r 's). We also have I_{in} connected with each II and each III , while I_{out} is connected with each II , and I_{in} and I_{out} are not connected. Finally, we have $z \in \{t, b\}$ indicating whether two II 's are connected —i.e., two II 's are adjacent iff they have the same value of z .

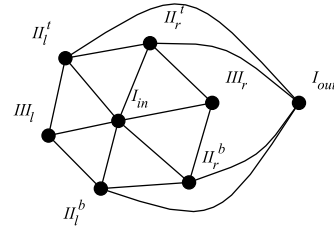


Fig. 7. Notation for the vertices of the graph used in the proof of Theorem 5.

To simplify our proof, in the following we use $h, k \in \{l, r\}$ where $h \neq k$ if not specified, and $i, j \in \{t, b\}$ where $i \neq j$ if not specified. Hence, for two vertices II_h^i and II_k^j we have:

- $h = k \Leftrightarrow$ the vertices are adjacent to the same III .
- $i = j \Leftrightarrow$ the vertices are adjacent

To prove that every possible labeling is jumping, we first temporary remove the two vertices of type I . Their removal produces a cycle composed of six vertices. Observe that a cycle has a non-jumping representation if we fix the label of one vertex and the other vertices are labeled in sequence, or if the labels of two adjacent vertices are swapped, i.e., the labels of any pair of adjacent vertices are at distance of at most two. Thus, for our cycle the feasible (i.e., non-jumping) sequences are:

- $III_h II_h^i II_k^i III_k II_k^j II_h^j$,
- $II_h^j III_h II_h^i II_k^i III_k II_k^j$,
- $II_k^j II_h^j III_h II_h^i II_k^i III_k$,

as well as all sequences obtained by permuting exactly two consecutive vertices. (For the sake of brevity, in the following we consider only the sequences without such a

permutation. However, this proof can easily be extended to accommodate the permuted sequences.)

Before adding the two I 's to the sequences given above, we consider the following infeasible, i.e. jumping, configuration. Note that we use the notation (\dots) to signify to any vertex or sequence of vertices:

$$(\dots)II(\dots)I(\dots)I(\dots)II(\dots) \quad (1)$$

In general, the two I 's cannot be placed between any two pair of vertices of type II , since the I 's are not connected by an edge, but must be connected to every II .

From the infeasible configuration 1, it follows that at least one I should be placed to the left (or right) of all the II 's. Without loss of generality, we consider only placing I to the left of the II 's. After placing one I in this way, we have the following possibilities:

- $I III_h II_h^i II_k^i III_k II_k^j II_h^j$
- $III_h I II_h^i II_k^i III_k II_k^j II_h^j$
- $I II_h^j III_h II_h^i II_k^i III_k II_k^j$
- $I II_k^j II_h^j III_h II_h^i II_k^i III_k$

We now observe that two nonadjacent II 's cannot be placed between the two I 's, that is,

$$(\dots)I(\dots)II^h(\dots)II^k(\dots)I(\dots) \quad (2)$$

is unfeasible if $h \neq k$ is infeasible. This is because there is no an edge between II^h and II^k , while there must be an edge between each I and each II . From this infeasible configuration, it follows that between the two I 's there can be only zero, one, or two II 's.

Because III_k and II_h do not share an edge, the sequence

$$(\dots)I(\dots)III_k(\dots)II_h(\dots)II_k(\dots) \quad (3)$$

is infeasible, and because II_k^i and II_k^j do not share an edge, we also have

$$(\dots)I(\dots)II_k(\dots)II_k(\dots)III_k(\dots) \quad (4)$$

is infeasible. As such, we can eliminate the first and the last sequences in the previous list.

The following list reports all the possible sequences with both I 's. Here, the available positions for the second I that remain after considering the previous infeasible configurations are shown inside parentheses:

- $III_h I (I) II_h^i (I) II_k^i (I) III_k (I) II_k^j II_h^j$
- $I (I) II_h^j (I) III_h (I) II_h^i II_k^i III_k II_k^j$

Now, if we identify the two I 's as I_{in} and I_{out} , we find that the following are infeasible configurations:

$$(\dots)I_{in}(\dots)I_{out}(\dots)III(\dots)II(\dots) \quad (5)$$

because I_{out} and III do not share an edge, and

$$(\dots)II(\dots)III_h(\dots)I_{out}(\dots)II_h(\dots) \quad (6)$$

because no III and I_{out} share an edge. Thus, II cannot be followed by a III , a I_{out} and the adjacent II of the previous III .

We observe that the previous configurations can occur in any of the two sequences. So the leftmost I can only be I_{out} :

- $III_h I_{out} (I_{in}) II_h^i (I_{in}) II_k^i (I_{in}) III_k (I_{in}) II_k^j II_h^j$
- $I_{out} (I_{in}) II_h^j (I_{in}) III_h (I_{in}) II_h^i (I_{in}) II_k^i III_k II_k^j$

We finally conclude that all the remaining sequences are jumping since they each contain at least one of the infeasible configurations:

$$(\dots)III(\dots)I_{out}(\dots)I_{in}(\dots)II(\dots) \quad (7)$$

because I_{out} and I_{in} do not share an edge, and

$$(\dots)I(\dots)II_k^i(\dots)II_h^i(\dots)III_k(\dots) \quad (8)$$

because II_k^i and II_h^i do not share an edge

It follows that the graph does not have a non-jumping labeling. As mentioned, this proof can easily be extended to any exchange of an adjacent pair of vertices of the cycle.

Proof of Theorem 10:

Theorem. *Every 4-leaf power graph admits an L-embedding.*

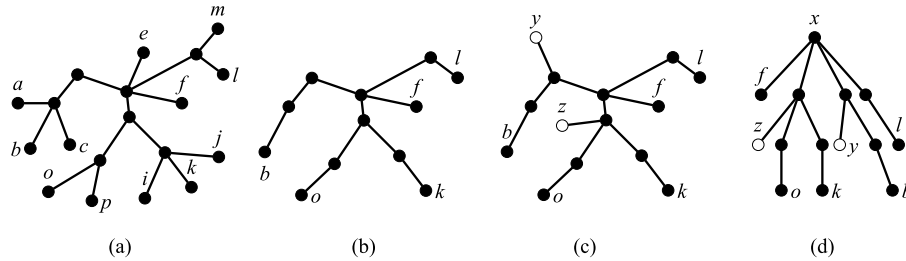


Fig. 8. (a) A tree T , which is modified by (b) removing multiple siblings, and (c) adding dummy leaves to the internal vertices. (d) A rooted simplified leaf-tree of T .

Before we begin our proof, we first define some notation and terminology. Let $G = (V, E)$ be a 4-leaf power graph. Then there is a tree T' whose leaves correspond to the vertices of G in such a way that two vertices are adjacent in G precisely when their distance in T' is at most 4. Let v be a vertex of G . We denote the graph obtained by removing v and all edges incident to v by $G - v$.

To make T' as a simpler and more uniform tree, we remove and add some leaves on T' as follows. We first remove siblings leaves from T' , i.e., leaves that have the same parent, and add new leaves to each internal node without any child-leaf to create a *rooted simplified leaf-tree* T of G from T' , as follows. Let u and v be two siblings leaves in T' . Then $N(u) = N(v)$ in G and $(u, v) \in E$. Hence, u is a true twin of v in G . According to Theorem 9, if we have $L(G - v)$ then we have $L(G)$. For each group of sibling leaves, we keep exactly one leaf, removing the others. For example, Figure 8(b) shows the transformed tree after removing multiple siblings from the tree shown in Fig. 8(a). We now add a dummy leaf to the internal vertices of T that do not have a child-leaf. In Figure 8(c), vertices y and z are dummy leaves. The dummy vertices will be removed from $L(G)$ after the construction of L-embedding of G . We make T a rooted tree by selecting an arbitrary internal vertex as its root (see Fig. 8(d)). Observe that every internal vertex has exactly one leaf in the rooted simplified leaf-tree.

Let v be a vertex of T . We denote the subtree rooted at v by T_v . We denote the parent of v by v' , and the parent of v' by v'' . Vertices u and v are said to be *siblings* if $u' = v'$. A vertex u is said to be an *uncle* of v if $u' = v''$. A vertex u is said to be a *p-uncle* of v if u' is parent of v'' . Vertices u and v are said to be *cousins* if $u'' = v''$. A vertex u is said to be a *nephew* of v if $u'' = v'$. The length of the largest path from the root of T to any leaf is called *depth* of T .

Since G is a 4-leaf power graph, then $(u, v) \in E$ if and only if one of the following is true:

1. u and v are siblings,
2. u and v are cousins,
3. u is an uncle of v or v is an uncle of u , or
4. u is a p-uncle of v or v is a p-uncle of u .

By definition, a leaf u has exactly one uncle and exactly one p-uncle. Also, cousins have a common uncle and a common p-uncle.

We now draw each group of cousins as a *fully connected L-embedding* in which all of the L-segments intersect one another (see Fig. 9).

Let r be the root of T . Since T is a rooted simplified leaf-tree, r has exactly one child-leaf, which we call r_c . We denote a nephew of r_c by r_{cc} , and the subtree induced by the vertices r , r_c , and r_{cc} by T_{conf} .

We maintain two configurations, a "*Rectangle-configuration*" and an "*L-configuration*," for the drawing of leaves of T_{conf} of T . The properties of a *Rectangle-configuration* are:

RconPro1 The horizontal arms of cousins intersect the vertical arm of their uncle, maintaining a subdivided L-shaped free region for each cousin

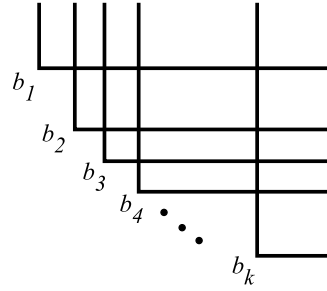


Fig. 9. A fully connected L-embedding

RconPro2 The vertical part of each subdivided L-shaped free region is visible from the horizontal arm of corresponding cousin

RconPro3 The horizontal part of every subdivided L-shaped free region is visible from the vertical arm of the uncle

The properties of an *L-configuration* are the following:

LconPro1 The vertical segments of cousins intersect the horizontal segment of their uncle, maintaining a subdivided rectangle shape-free region for each cousin

LconPro2 The right part of each subdivided rectangle shape-free region is visible from the vertical arm of its corresponding cousin

LconPro3 The top part of every subdivided rectangle shape-free region is visible from the horizontal arm of the uncle

Figure 10(f) depicts a Rectangle-configuration for the tree drawn in Fig. 10(d), and Fig. 10(e) depicts an L-configuration for the tree drawn in Fig. 10(d).

We need the following lemma for the proof of Theorem 10

Lemma 4. *Let G be a 4-leaf power graph and T be a corresponding rooted simplified leaf-tree of G . Then T_{conf} admits a Rectangle-configuration and an L-configuration.*

Proof. Let r be the root of T . Assume that T_{conf} of T contains $k + 1$ leaves. These are r_c and k nephews of r_c (for some $k \geq 1$); the k nephews are cousins of each other. We find a Rectangle-configuration of T_{conf} as follows. We take a fully connected L-embedding of the k cousins and add $L(r_c)$ such that the horizontal arms of the k cousins intersect the vertical arm of $L(r_c)$. It is easy to verify that the properties of the Rectangle-configuration hold (see Fig. 10(b)). Similarly, we find L-configuration of T as follows. We take a fully connected L-embedding of k cousins and add $L(r_c)$ such that the vertical arms of the k cousins intersect the horizontal segment of $L(r_c)$. This drawing maintains the properties of L-configuration of T (see Fig. 10(c)).

Proof. We now begin a proof of the main theorem by induction. Let G_ρ be a 4-leaf power graph and T be a corresponding rooted simplified leaf-tree of G_ρ with depth ρ . Let r be the root of T . We claim that G_ρ admits two L-embedding such that T_{conf} admits the Rectangle-configuration in one L-embedding and the L-configuration in the other L-embedding. Our induction is based on depth of T .

Base case: The depth $\rho = 2$. Let G_2 be a 4-leaf power graph and T be a corresponding rooted simplified leaf-tree of G_2 with depth 2. Since $T_{conf} = T$, by Lemma 4 G_2 admits two L-embeddings such that T_{conf} admits the Rectangle-configuration in one L-embedding and the L-configuration in the other L-embedding (see Fig. 10(a)-(b)).

Induction case: We now assume that our claim holds for any depth $\rho - 1$. For the inductive step, we have to prove that it holds for depth ρ .

Let G_ρ be a 4-leaf power graph and T be a corresponding rooted simplified leaf-tree with depth ρ of G_ρ . Let r be the root of T (Fig. 10(a)). By Lemma 4, the drawing of T_{conf} maintains the properties of a Rectangle-configuration, but we need to show how to place $L(T_{r_{cc}})$ into the L-shaped free region between $L(r_{cc})$ and $L(r_c)$. (Any L-embedding consists of horizontal and vertical line segments so width or height or both of the drawing can be resized.)

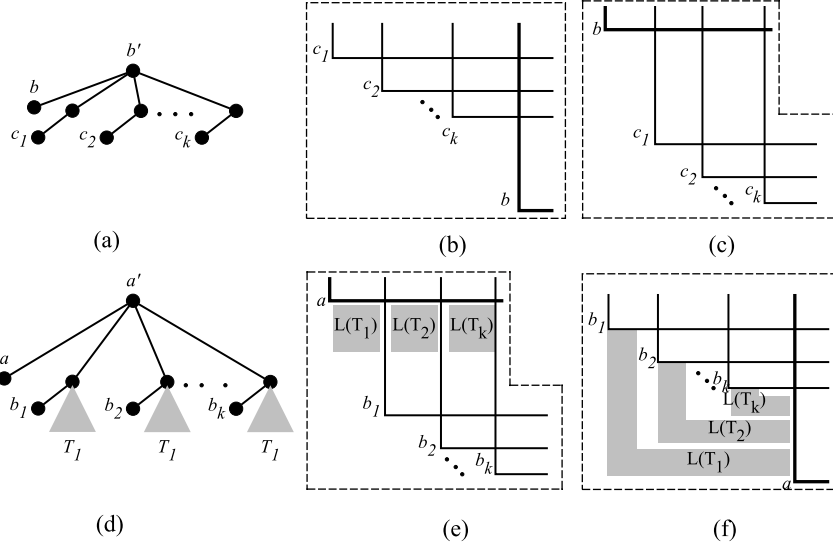


Fig. 10. (a) The tree T used for the base case, (b) a Rectangle-configuration of T , and (c) an L-configuration of T . (d) The tree $T_{a'}$ used in the induction step, (e) an L-configuration of $T_{a'}$, and (f) a Rectangle-configuration of $T_{a'}$.

Let ρ' be the depth of $T_{r'_{cc}}$. Since $\rho' \leq \rho - 1$, by the inductive hypothesis, $G_{\rho'}$ has an L-configuration with respect to the root r_{cc} of $T_{r'_{cc}}$ with properties LconPro2 and LconPro3. Note that r_{cc} of T and r'_{cc} of $T_{r'_{cc}}$ are the same vertices in T . We therefore place the corner of $L(r'_{cc})$ on the corner of $L(r_{cc})$ to make one L-segment. We know r_c is the p-uncle of each r'_{cc} . Hence $L(r_c)$ and $L(r'_{cc})$ should cross. The crossing between the nephews of r_{cc} in $T_{r'_{cc}}$ with r_c in T can be achieved by extending the horizontal arm of $L(r'_{cc})$ because of LconPro2. Let a and b be two nephews of r_c . Then the distance between any leaf in T_a and any leaf in T_b is greater than 4, except for a and b . Thus G_{ρ} admits an L-embedding such that T_{conf} admits the Rectangle-configuration.

We now show that G_{ρ} admits an L-embedding such that T_{conf} admits the L-configuration. By Lemma 4, the drawing of T_{conf} maintains the properties of L-configuration but we need to show how to place $L(T_{r'_{cc}})$ into the L-shaped free region between $L(r_{cc})$ and $L(r_c)$.

Let ρ' be the depth of $T_{r'_{cc}}$. Since $\rho' \leq \rho - 1$, by the inductive hypothesis we know that $G_{\rho'}$ has an Rectangle-configuration with respect to the root r_{cc} of $T_{r'_{cc}}$ with properties RconPro2 and RconPro3. Note that r_{cc} of T and r'_{cc} of $T_{r'_{cc}}$ are the same vertex. We place the corner of $L(r'_{cc})$ on the corner of $L(r_{cc})$ to make one L-segment. We know r_c is the p-uncle of each r'_{cc} . Hence $L(r_c)$ and $L(r'_{cc})$ should cross. The crossing between the nephews of r_{cc} in $T_{r'_{cc}}$ to r_c in T can be done by extending vertical line segment of $L(r'_{cc})$ because of LconPro2. Let a and b be two nephews of r_c . Then the distance between any leaf in T_a and any leaf in T_b is greater than 4, except for a and b . \square

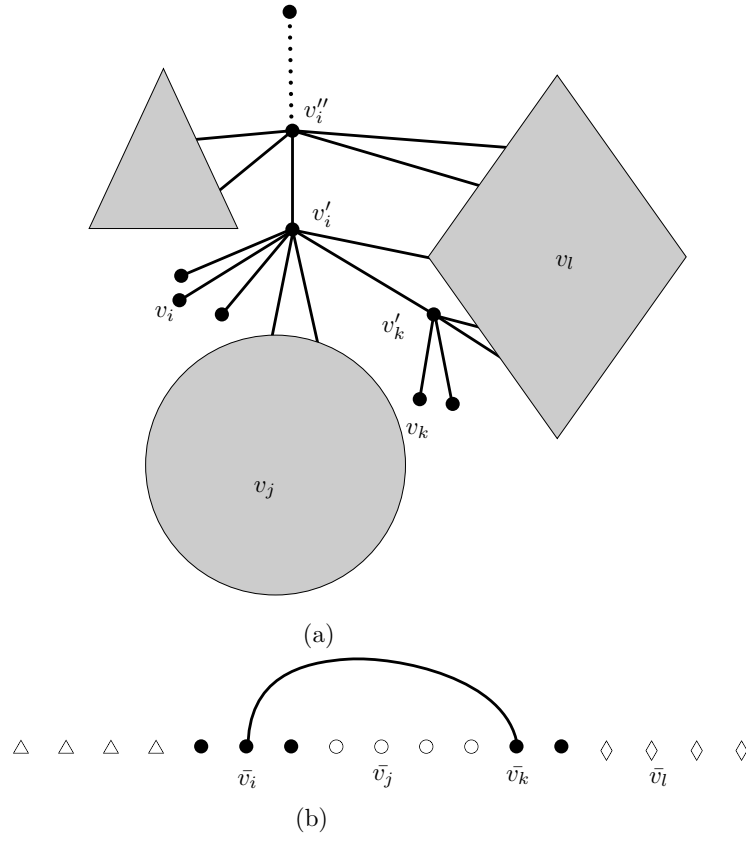


Fig. 11. Illustration for the proof of Theorem 2.